بحث عن المتحكم الصغير
PIC16F84

مقدمة عن للمتحكم الصغير

المتحكم الصغري أو المايكروكنترلر هو في الواقع كمبيوتر صغيرمصمم خصيصا ليقوم بأعمال معينة. ويستخدم الذاكرة لتخزين الأوامر المبرمجة و القيام بتنفيذ هذه الأوامر مثل التشغيل والاطفاء ، التوقيت ، العد ، الحساب وغير ذلك من العمليات.

تم استعمال أول مايكروكنترولر في عام ١٩٦٩ ومنذ ذلك الوقت بدأت هذه المتحكمات بالانتشار حتى بات من الصعب العمل في مجال الاليكترونيات الحديثة بدون معرفة الميكروكنترولر.

وهذه المتحكمات العجيبة موجودة في داخل العديد من الأجهزة التي نستخدمها في حياتنا اليومية. فمثلاً في السيارة نجد أن الفرامل (الكوابح) و مثبت السرعة يتم التحكم فيها عن طريق المايكروكنترولر.

ولو نظرنا إلى فرن المايكروويف في المطبخ لوجدنا بداخله مايكروكنترولر للتحكم بالتوقيت والحرارة بحسب الخيارات التى نطلبها عند الطبخ.

والأمثلة على الأجهزة التي يوجد بداخلها ميكروكنترولر كثيرة منها الهواتف الجوالة ، الثلاجات ، الغسالات ، التلفزيونات ، كاميرات الفيديو ، الكاميرات الرقمية وغير ذلك كثير.

خواص المايكروكنترولر :

كما ذكرنا فإن المايكروكنترولر هو عبارة عن كمبيوتر صغير ولكن ما هي الخواص التي تميز المايكروكنترولر عن باقي الكمبيوترات ؟

أهم هذه الخواص هي:

1- يكون المايكروكنترولر عادة بداخل جهاز آخر للتحكم بذلك الجهاز كما ذكرنا سابقاً.

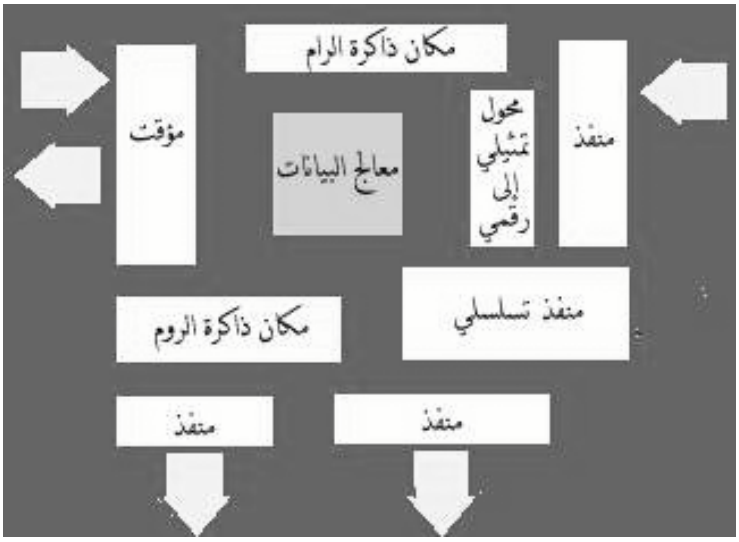
2- يكون في المايكروكنترولر مايحتاجه من الذاكرة مثل الرام والروم (RAM & ROM) فهو ليس بحاجة إلى شرائح خارجية للذاكرة.

3- يكون عمل المايكروكنترولر محدد بمهمة واحدة وتنفيذ الأوامر في برنامج واحد يكون مخزناً في ذاكرة المايكروكنترولر.

4- يكون استهلاك المايكروكنترولر من الطاقة صغيراً جدا بالنسبة للكمبيوترات الأخرى فمثلا بعضها يستهلك ٥٠ ميللي وات بينما الكمبيوتر العادي الذي نستخدمه في منازلنا قد يستهلك ٥٠ وات.

مكونات المايكروكنترولر :

تحتوي شريحة المايكروكنترولر كما هو موضح في الصورة على معالج بيانات ، ذاكرة رام و ذاكرة روم ، منافذ للمداخل والمخارج (I/O interfaces) ، مؤقتات وانظمة أخرى مثل محولات القيم التمثيلية إلى رقمية (ADC).



والآن لنعطي نبذة عن عمل كل من هذه الأجزاء:

معالج البيانات:

و يعتبر قلب المايكروكنترولر ويختلف باختلاف الجهاز المستخدم وكذلك مصنع الجهاز فمثلاً معالج البيانات المستخدم في جهاز الهاتف الجوال يختلف عن ذلك المستخدم في فرن المايكروويف.

الذاكرة:

وتنقسم إلى رام و روم. أما الرام فتستخدم لتخزين المعلومات ويتراوح حجمها بين ٢٥ بايت و ٤ كيلوبايت بحسب المايكروكنترولر.

أما الروم فيتراوح حجمها بين ٥١٢ بايت و ٤٠٩٦ بايت وقد يصل حجمها إلى ١٢٨ كيلوبايت في بعض المايكروكنترولات. وتستخدم الروم لتخزين البرامج التي تحتوي الأوامر التي ينفذها المايكروكنترولر.

) حيث يمكنROMوذاكرة الروم قد تكون من نوع الروم (برمجتها مرة واحدة فقط وقد تكون من نوع إي بروم)EEPRPM) أو إي إي بروم (EPROM(

Flash Technologyو يقال أنها تدعم تقنية

أي يمكن برمجتها عدة مرات يصل إلي ألف مرة .

منافذ المداخل والمخارج:

وهذه المنافذ الرقمية توفر للمايكروكنترولر الطريق للتعامل مع العالم الخارجي و الاجهزة الخارجية. حيث يمكن استعمالها لتشغيل الدايودات المضيئة والمرحلات

ويختلف عدد هذه المنافذ بحسب المايكروكنترولر

المنفذ التسلسلي:

المنفذ التسلسلي يسمح بتبادل المعلومات بين المايكروكنترولر و الأجهزة الأخرى مثل الكمبيوتر و المايكروكنترولات الأخرى.

المؤقت:

يسمح للمايكروكنترولر بالقيام بالمهمات لفترات زمنية محددة

المحول التمثلي إلى رقمي:

)Analogو هو يترجم المعلومات الداخلة بالهيئة التمثيلية ( ) حتى يتمكن المايكروكنترولر منDigitalإلى هيئة رقمية (فهمها والاستجابة لها.

برمجة المايكروكنترولر :

توفر الشركات المصنعة العديد من أنواع المايكروكنترولات للمحترفين والهواة حيث يمكن عمل التجارب المختلفة عليها.

هذه المايكروكنتولات يمكنها القيام بمهام مختلفة بحسب الأوامر التي تعطى لها وهذه الأوامر تسمى بالبرنامج. فبامكان الشخص تغيير العمل الذي يقوم به المايكروكنترولر بتغيير هذه الأوامر في البرنامج.

كتابة البرنامج تحتاج من الشخص إلى معرفة جيدة بلغات البرمجة مثل الأسمبلي أو غيرها من اللغات.

لتغيير البرنامج في المايكروكنترولر سيحتاج الشخص إلى جهاز وسيط لتحميل البرنامج الجديد من الكمبيوتر الشخصي إلى المايكروكنترولر. حيث يركب فيه المايكروكنترولر لتحميل البرنامج فيه ثم بعد

ذلك يزال المايكروكنترولر إلى الدائرة التي سوف يستخدم فيها.

ما هو الـ
PIC16F84

هو عبارة عن دائرة متكامله صنعت بواسطة شركة www.microchip.com
) هو مثل الميكروبرسسورMicroprocessor (
) ولكن الميكروكنترولر فيه إضافات علىMicroprocessor(
الميكروبرسسور وهي انه له ذاكرته الخاصة بالداخل والتي تستخدم لتخزين البرنامج بها، وكذلك يحتوي على ذاكرة المعلومات والتي تستخدم لتخزين المتغيرات، وأيضا يحتوي علىPIC. مداخل وفي نفس الوقت هي مخارج للـ

والفرق بين الميكروكنترولر والميكروبرسسور هو أن الثاني يحتاج الى إضافات عديده لكي يعمل ويقرأ البرنامج المعد لهIC، وأما الميكروكنترولر فلا يحتاج الى اي شئ فقط الـ وكرستال لكي يعمل ويقرأ البرنامج الذي بداخله.

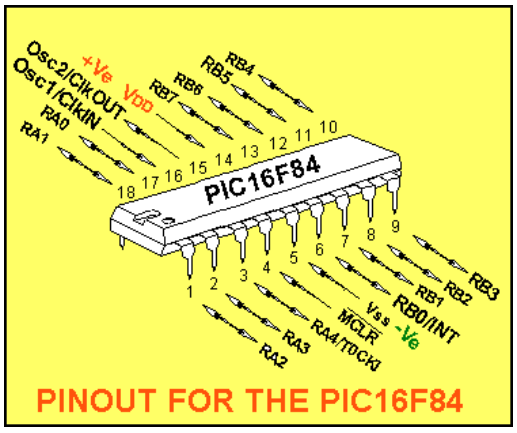
جعلتنا نختارها علىPIC16F84aميزه لطيفه ورائعه في الـ بداخلها وهذا يعني انناEEPROMغيرها وهي وجود ذاكرة نستطيع أن نبرمجها ونمسح البرنامج في لحظات ولأكثر من مره ولن نحتاج الى الأشعه فوق البنفسجية لمسح البرنامج.

مميزات المتحكم الصغير PIC16F84

1 - تحتوي على ٣٥ أمر برمجة
٢Single Cycle - كل الأوامر دورة واحدة (ما عدا أوامر (Two Cycle) فهي دورتان (Branches القفز)
٣Input/Output - ١٣ مدخل ومخرج في نفس الوقت
٤ - تحتوي على نظام الحفاظ على إستهلاك الطاقة (Power Saving)
٥ - محمي ضد الكهرباء الستاتيكية

الدائرة المتكاملة للمتحكم الصغير PIC16F84

شكل الدائرة المتكاملة من الخارج



مخطط أرجل الدائرة المتكاملة للمتحكم الدقيق

جدول يوضح أرجل المتحكم و وظيفو كل منها :

رقم الرجل	
1	RA2 - Port A
2	RA3 - Port A
3	RA4/TOCK1 – Port A
4	MCLR - Master clear input (active low)
5	Vss – أرضي
6	RB0/INT - Port B
7	RB1 - Port B
8	RB2 - Port B
9	RB3 - Port B
10	RB4 - Port B
11	RB5 - Port B
12	RB6 - Port B
13	RB7 - Port B
14	Vdd – الجهد الموجب
15	OSC2/CLKOUT - خرج المذبذب
16	OSC1/CLKIN – دخل المذبذب
17	RA0 - Port A
18	RA1 - Port A

PIC16F84
Memory Organization
توزيع الذاكرة

يتكون المتحكم الدقيق من نوعين من الذاكرة هما :

1. ذاكرة البرنامج (Program Memory)

word و تسع ١٠٢٤ مكان مكون من
1 word = 14 bits

2. ذاكرة البيانات (Data Memory)

و بها أثنان من الذاكرة

أولاً - ذاكرة الوصول العشوائي

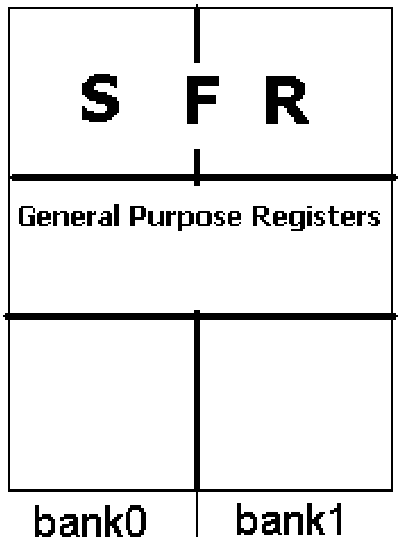
و تم تقسيمها رأسياً إلي

1. SFR(Special Functoin Register)
مسجلات الوظائف الخاصة

2 . GPR(General Purpose Registers)
مسجلات الأغراض العامة

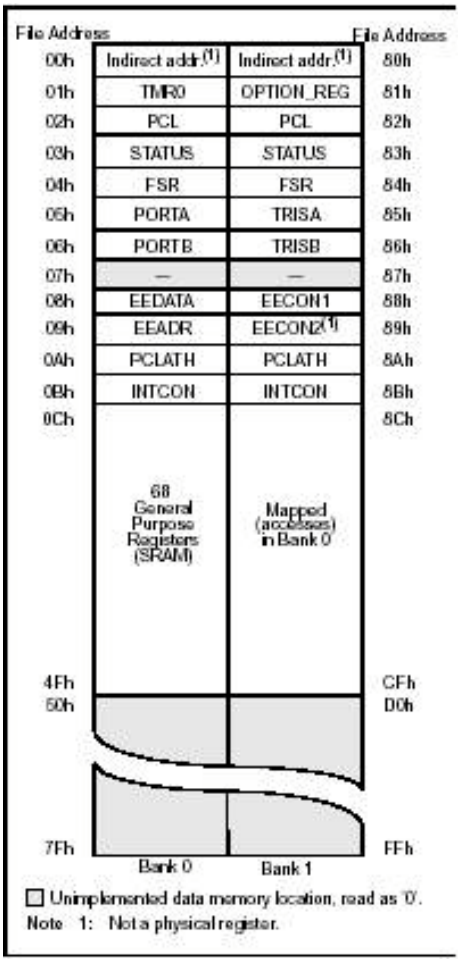
و أفقياً إلي

1. bank0
2. bank1



و سوف نتحدث عن كل منهما من خلال شرحنا لأمثلة عملية علي أستخدم المايكروكنترولر في الدروس القادمة

و هذا مخطط تفصيلي لمحتوي ذاكرة الوصول العشوائي للمتحكم الدقيق :



ثانياً -
ذاكرة القراءة فقط القابلة للبرمجة و المحو الكهربائي EEPROM(Electrisity Erisable Programable ROM)

و هي ذاكرة يمكن إستخدامها من قبل البرنامج عمليات القرءة و الكتابة) و الموجود بداخل ذاكرة البرنامج فمثلاً عند التحكم في درجة حرارة غرفة فتتم برمجة النظام (لتحديد درجة الحرارة المطلوبة) من قبل مستخدم النظام (عن خاصة لذلك) فذلك يتطلب ذاكرةKeybadطريق إدخالها من EEPROM
لأمكانيتها علي الأحتفاظ بمحتوياتها (الأحتفاظ بدرجة الحرارة المطلوب ضبط الحرارة عليها)بعد فصل التيار الكهربي عنها
تسع هذة الذاكرة إلي ٦٤ مكان بسعة واحد بايت يعني ثماني بت

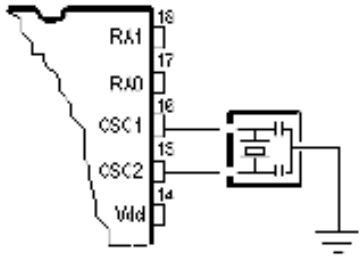
مذبذب المتحكم الصغير
Oscillator

المذبذب :

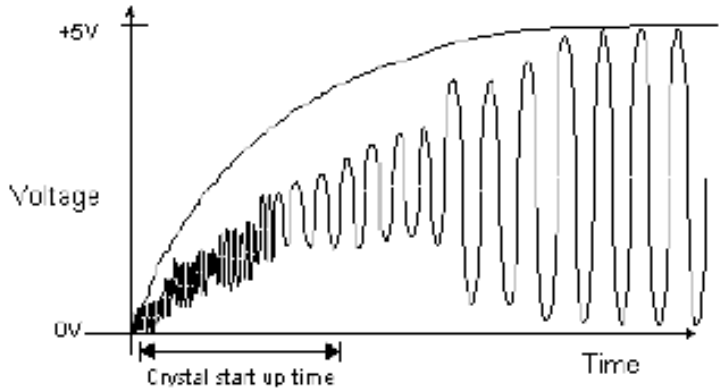
هو الذي يقوم بتوليد نبضات الساعة لتشغيل المعالج الدقيق الموجود بداخل المتحكم الصغير و هناك نوعان من المذبذبات المستخدمة مع المتحكم الصغير

النوع الأول : مذبذب مقاومة مكثفRC-Oscillator

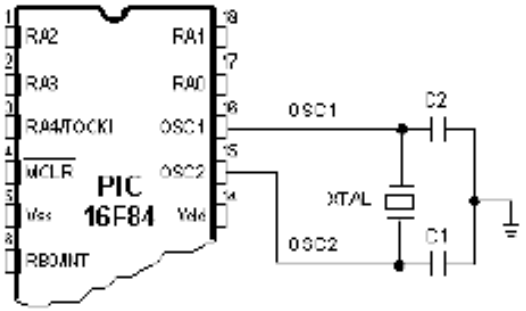
و يوصل كما بالشكل



و من عيوبة أنة ليس دقيق فبمجرد توصيل مصدر التغذية لن يعمل بالسرعة المرجوة منة فهو يأخذ وقت يسمي وقت بداية المذبذب حتي يعمل حيث تتأثر نبضاتة بمعدل تغير الجهد و شدة التيار
و الشكل التالي يوضح كيفية تأثير التيار علية و زمن وقت التشغيل الاول



النوع الثاني : مذبذب كرستالة
و توصل كرستالة التوقيت هذة بالمتحكم الصغير كما بالشكل



و تتميز بالدقة عن مذبذب مقاومة مكثف السابق

الدخل والخرج للمتحكم الدقيق

PIC 16F84

يتعامل المتحكم الدقيق مع العالم الخارجى عن طريق منافذ الدخل والخرج الموجودة فيهما

وهما اثنان :

أولاً المنفذ أ

PORTA

يتكون من خمسة بت
يمتلك خمسة ارجل من الدائرة المتكاملة PORTA
وتكون بهذا الشكلPic 16f84

RA4	RA3	RA2	RA1	RA0
0	0	0	0	0

ثانياً المنفذ ب

PORTB

يتكون من ثمانية بت
يمتلك ثمانيةارجل من الدائرة المتكاملة PORTB
وتكون بهذا الشكلPic 16f84

RB7	RB6	RB5	RB4	RB3	RB2	RB2	RB1	RB0
0	0	0	0	0	0	0	0	0

ويلزمنا للتعامل مع منافذ الدخل والخرج

اولآ

اختيار اى من المنافذ نريد التعامل مع العالم الخارجى عن طريقه

ثانيآ

اعداد ارجل المنفذ بمعنى تحديد كل رجل ما اذا كانت دخل او خرج .

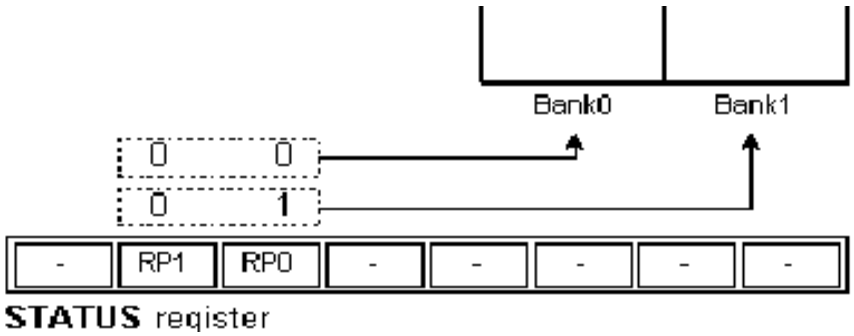
سنقوم الان بتحديد الدخل والخرج للـ
PORTB
لتحديد الأرجل الدخل و الخرج للمنفذ ب يلزمنا وضع نموذج لبتات البمنفذ في مشجل مسمي
TRISB
فمثلا نريد أن تكون أرجل المنفذ ب خرج كلها فنضع في المسجل السابق هذا النموذج
00000000

تعنى ان الرجل يكون دخل 1--<
تعنى ان الرجل يكون خرج 0--<

و لكن اهذا المسجل موجود في البانك صفر فيلزمنا أولاً الأنتقال إلي هذا البانك قبل التعامل مع هذا المسجل

STATUS Registerشرح مسجل الحالة

لكى نقوم بالذهاب الى البانك صفر يكون ذلك عن طريق مسجل الحالة يتكون اى مسجل من ثماني بتات كل بت فى اى مسجل له وظيفة خاصة وسنقوم بالتعرف على الـ البتابت الخاص بالتنقل من و إلي البانك واحد و البانك صفر البت رقم رقم خمسة و يسمى فى مسجل الحالة RP0 هو الاكثر اهمية حيث انه يستخدم للذهاب الى البانك واحد و البانك صفر فاذا كانتقيمتة تساوي واحد يكون عندالبانك واحد ،وعندما يساوىصفر فينتقل الى البانك صفر كما هو موضح بالشكل الاتى



نقوم بتكملة المثال نريد تحميل هذا النموذج المراد اعداده للمنفذب فتكون التعليمات كما يلى

```
BSF     STATUS,RP0
MOVLW   b'000001100'
MOVWF   TRISA
```

كيفية الكتابة الى المنافذ

نفترض ان لدينا ثنائى مضئ موصل على البت رقم صفر في المنفذ أ ونريد اضائته نكتب الاتى

BSF PORTA,0

ولكن يجب هنا مراعاة ان مسجل الهدف منفذ أ يوجد فى البانك صفر - كما وضحنا سابقآ - لذلك فيجب الانتقال الى البانك صفر
عن طريق البت رقم خمسة في مسجل الحالة - كما وضحنا سابقا - فيمكن الانتقال الى المسجل عن طريق التعليمة

BCF STATUS,5

واذا اردنا اطفاء الثنائى فنكتب الامر التالى

BCF PORTA,0

كيفية القراءة من المنافذ

نفترض اننا قمنا بتعريف البت رقم صفر فى المنفذ أ على انه دخل والبت رقم صفر في المنفذ ب على
انها خرج وفكرة عمل البرنامج هى عند الضغط على الزر الموصل بالبت رقم صفر فى المنفذ أ تتم
أضاءة الثنائى المضئ الموصل بالبت رقم صفر فى المنفذ ب وسوف نستخدم التعليمة
(BTFSC) .

المقاطعة
Interrupts

وهى الاشارة التى توقف عمل المتحكم الدقيق او المعالج الدقيق بسبب حدوث شئ ما ؛ كمثال حدوث مقاطعة كل ساعة لتنفيذ مجموعة معينة من التعليمات ؛ مثال اخر جرس التليفون فعندما يرن جرس التليفون يجب مقاطعة حديثك مع الاخرين ورفع سماعة الهاتف والتحدث وعند الانتهاء من المحادثة يجب مواصلة الحديث مع الاخرين وهنا نعتبر ان عملية رفع السماعة واجراء المحادثة التليفونية وعملية العودة والتحدث مع الاخرين هى عبارة عن اجراء مقاطعة الفرعى

Interrup roution

يعتبر المثال السابق مثال حى على كيفية تأثير المقاطعة على المعالج الدقيق او المتحكم الدقيق ولكنفى مخطط سير البرنامج توجد بعض الوظائف فى فى الدائرة .
توصل الماطعات الى المتحكم الدقيق ولكن عند حدوثه يخرج البرنامج الرئيسى عن عمله ويذهب الى تنفيذ المقاطعة الفرعيةوعند الانتهاء من اجراء المقاطعة الفرعية يرجع مرة اخرى البرنامج الئيسى الى متابعة سير البرنامج .

يمتلك المتحكم الدقيق اربع مصادر للمقاطعة يمكن تقسيمهم الى نوعين من المقطعات

النوع الاول: مقطعات خارجية

نرىان الرجل رقم ستة للمتحكم الدقيق معنونة بأسم RB0/INT
فالرمز
RB0
بالطبع البت رقم صفر من المسجل المنفذ ب والرمز
INT
رمز الى انه يمكن ضبط اعدادات المتحكم الدقيق ان تكون هذه الرجل مقاطعة خارجية

هى الرجول
13,12,11,10
وهى تمثل البتات من خمسة إلي سبعة من المسجل
المنفذ ب هما الاخرين يمكن اعدادهم
كمقاطعة خارجية

قبل ان يمكننا استخدام المقاطعات سواء الت رقم ستة او ارجل المسجل المنفذ ب نحتاج الى شيئين

<u>الاول</u>
هواخبار التحكم الدقيق اننا سوف نستخدم المقاطعات
<u>الثانى</u>
نحتاج التحديد اى مقاطعات سوف نستخدم سواءالبت رقم ستة او ارجل المسجل المنفذ ب

النوع الثاني : مقاطعات داخلية

مثل
المؤقت صفر

TMR0

يمتلك المتحكم الدقيق مسجل واحد للعد و يتم ضبط سرعة نسبتا لسرعة الساعة للمتحكم
سرعة نبضات الساعة) عن طريق ثلاثة بتات موجودين في) مسجل الخيارات

OPTION Register

$\overline{RBPU}$	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0

bit7

و هما
PS0
PS1
PS2
و يتم عن طريقهما تحديد النسبة بين نبضات كرستالة التوقيت و سرعة المؤقت صفر TMR0

و يتم ضبط الثلاث بنات المذكورة حسب الجدول التالي

Bits	TMR0
000	1 : 2
001	1 : 4
010	1 : 8
011	1 : 16
100	1 : 32
101	1 : 64
110	1 : 128
111	1 : 256

حيث عندما يكون القيم المنسوبة للبتات الثلاث المذكورة هي ١١١
فأن النسبة بين نبضات الساعة و سرعة الـ TMR0
هي ١ : ٢٥٦
أي كل ٢٥٦ نبضة من نبضات الساعة التي يعمل بها المتحكم الصغير يزيد المؤقت صفر المؤقت صفر بقيمة واحد

كيفية الأستدلال أن المؤقت قد أمتلأ
Timer0 Overflow

يتم ذلك عن طريق البت رقم ٥ في مسجل المقاطعة

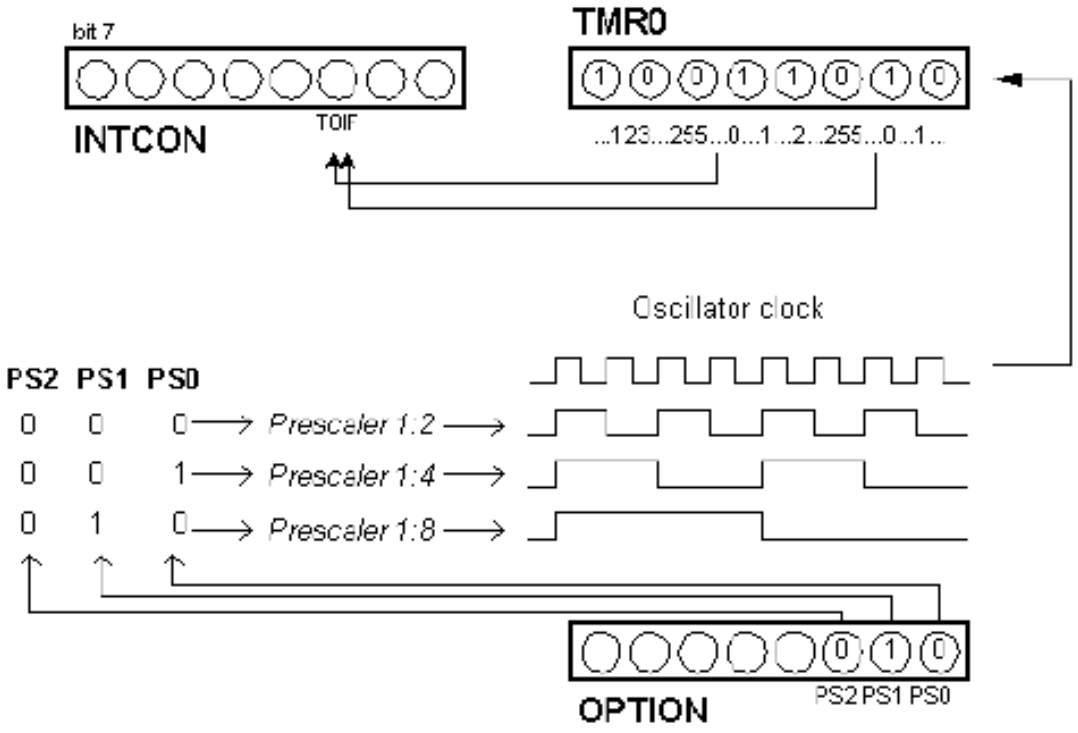
كما بالشكل . T0IEو المسماة

INTCON Register

GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

bit7

تكون قيمة هذا البت TMR0 للـ overflowفعند حدوث واحد و غير ذلك تكون قيمتة صفر



رسم يوضح العلاقة بين المسجلات الثلاثة

TMR0

OPTION

INTCON

كيفية كتابة برنامج المقاطعة:

عندما تصل الكهرباء الى المتحكم الدقيق يشير عداد البرنامج الى العنوان
(0000h)
وهذا العنوان يعتبر بداية ذاكرة البرنامج ويسمى
(Rest vector)
ولكن مهما كانت قيمة عداد البرنامج (عنوان التعليمة التى ينفذها المتحكم الدقيق حالياً)
عند حدوث مقاطعة يشير عداد البرنامج الى العنوان
(0004h)
ويسمىهذا العنوان
Interrupt vector
لذلك عند كتابة مثل هذه البرامج نحتاج ان نخبر المعالج الدقيق ان يقفز بعيداً عن العنوان
(0004h)
في بادية تنفيذ البرنامج اى يترك اجراء المقاطعة الذى يبدأ فى العنوان
(0004h)
و لا يقوم بتنفيذة من جراء نفسة بل عند حدوث المقاطعة
و نخبرة أننا نريد أستخدام المقاطعة عن طريق البن الأولي من مسجل التحكم في المقاطعات
INTCON Register

مجموعة التعليمات

Instruction Set

مجموعة التعليمات:

تتكون مجموعة التعليمات من

.1words=14bit

تنقسم هذه التعليمات الى ثلاث مجموعات.

المجموعة الاولى .

Byte-oriented file register operations.

التعليمة	d	مسجل الهدف(f)

d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address

تساوى الصفر فمعنى ذلك ان ناتجdاذاكانت قيمة ال) واذا كانتwالتعلية سوف يوضع داخل مسجل العمل (تساوى واحد فسوف يوضع الناتج فى نفس مسجل الهدف) .f(

المجموعة الثانية .

Bit-oriented file register operations.

التعليمة	B(bit)	f(مسجل الهدف)

f(داخل مسجل الهدف(bit هى رقم الbحيث ان ال
والمراد التعامل معها من خلال التعليمة .

المجموعة الثالثة .

Literal and control operations.

التعليمة	K

k(حيث ان الـ label. هو عبارة عن عنوان)

وفيما يلى
جدول توضيح لمجموعة الاختصارات:

F	وهو مسجل الهدف المراد التعامل معه
W	وهو مسجل العمل ومكانه بداخل المعالج الدقيق بداخل التحكم ، لك أن تتخيل أن هناك كمبيوتر داخل المتحكم الدقيق (ALU) و هذا الكمبيوتر يستخدم مسجل W مؤقت يدعي لمخاطبة أو التعامل مع منافذ الخرج أو خلافة و القيام بالعمليات الحسابية و المنطقية
K	وهو عبارة عن عنوان معين يشار اليه (Label)
d	يتم عن طريقها تحديد هدف التعليمة (المكان الذى سوف يوضع فيه ناتج التعليمة). فاذا كان الناتج صفر فان يخزن ( ) واذا كان واحد wفى مسجل العمل ( ).fفيخزن الناتج فى مسجل الهدف (
b	داخل مسجل الهدف المكونbitرقم ال bitمن ٨

التعليمة .

ADDLW

Add literal and w

الصيغة اللغوية :

ADDLW k

ووظيفتها:
(k) الى (wجمع محتوايات مسجل العمل)
).wووضع الناتج فى مسجل العمل(

مثال:

```
1. xc equ 0x5B
2. MOVLW 0x5
3. MOVWF xc
4. ADDLW xc
```

التعليمة

CLRW

CLEAR W

الصيغة اللغوية :
CLRW

ووظيفتها:
)fجمع ما بداخل مسجل العمل(

مثال :

اذا كان مسجل العمل به قيمة ولكن
w = 0x5
وبعد تنفيذ التعليمة
CLRW
فأصبح مسجل العمل خالى تماماً الان
w=0x00

التعليمة

Clear F

الصيغة اللغوية:

CLRF f

وهو يعمل على مسح محتوى مسجل الهدف (f) فى ال(bit) وانساب القيمة صفر الى كل f.

مثال:

اذاكان f = 01011101

CLRF f

f = 00000000

التعليمة

DECF

Decrement f

الصيغة اللغوية :

DECF f,d

أما هذه التعليمة هى انقاص قيمة مسجل الهدف (fالd)بواحد وهذا اذا كان قيمة الd تساوى واحد أما اذا كانت قيمتة تساوى صفر فأن قيمتة العمل (w)تظل ثابتة وتكون قيمة مسجل الهدف (f) مطروحآ منهمساوية لقيمة مسجل واحد.

مثال:

X= 0x9

w = 0x5

DECF X,1

X = 0x8

w = 0x5

X =0x9

w = 0x5

DECF X,0

X = 0x9

w = 0x8

التعليمة

INCF

Increment f

الصيغة اللغوية :

INCF f,d

أما هذه التعليمة هى زيادة قيمة مسجل الهدف (الfإذا كان قيمة هذا وهو واحد بd)تساوى تساوى صفر فأن قيمة الw)تظل ثابتة وتكون قيمة مسجل العمل ( ) مضاف اليهfمساوية لقيمة مسجل الهدف ( ) واحد.

مثال:

X= 0xff
w = 0x3
INCF X,1
X = 0x00
w = 0x3

X =0xff
w = 0x3
INCF X,0
X = 0xff
w = 0x00

التعليمة

MOVLW

Move literal to w

الصيغة اللغوية :

MOVLW k

والمقصود بهذه التعليمة هو تحريك محتويات (k ال) الى مسجل العمل (w)

مثال:

w = 0x5A

MOVLW 0x03

w = 0x03

التعليمة

OPTION

الصيغة اللغوية:

OPTION

) الىwوهى تحميل محتويات مسجل العمل (

OPTION_REGالمسجل

مثال :

قبل التعليمة **OPTION = '11111000'**

MOVLW b'00001110'

OPTION

OPTION ='00001110'

بعد التعليمة

التعليمة

MOVF

Move f

الصيغة اللغوية :
MOVF f,d

ووظيفتها:
تساوى الصفر فأن محتوياتdاذا كانت
) تذهب الى مسجل العملfمسجل الهدف (
تساوى واحد فأن محتوياتd) واذا كانت الwـ(
) تظل ثابتة .fمسجل الهدف (

مثال:

X = 0x03
w = 0x0A
MOVF X,1
X =0x0A
w =0x03

X = 0x03
w = 0x0A
MOVF X,0

X = 0x03
w = 0x03

التعليمة

MOVWF

Move w to f

الصيغة اللغوية :

MOVWF f

ووظيفتها :

هى تحميل محتويات مسجل الهدف (f)

الى مسجل العمل (w)

مثال:

```
OPTION_REG = 0xff
w = 0x4f
MOVWF   OPTION_REG
OPTION_REG = 0x4f
w = 0x4f
```

التعليمة

NOP

On operation

الصيغة اللغوية:

NOP

مثال:

PC = X

NOP

PC =X+1

حيث انPCهى Program Counter

التعليمة

BSF

bit set f

الصيغة اللغوية :
BSF f,b

) فىb) bitوهى انساب القيمة واحد لل)fمسجل الهدف(

مثال:
PORTB = 00000000
BSF PORTB,0
PORTB = 00000001

التعليمة

BCF

bit clear f

الصيغة اللغوية :

BCF f,b

) فى مسجلb) bitوهى انساب القيمة صفرلل الهدف ( )f

مثال:

PORTB = 00000001

BCF PORTB,0

PORTB = 00000000

التعليمة

RLF

Rotate left through carry

الصيغة اللغوية:

RLF f,d

ووظيفتها:

تدوير مسجل الهدف(f مقدار bit واحدة)
واذاكانت carry flag جهه اليسار عن طريق
(d تساوى صفر فأن الناتج فى مسجل)
(w واذا كانت تساوى واحد فأن العمل)
الناتج يوضع داخل مسجل الهدف.

مثال:

اذاكان

f = 00000000
RLF f,1
f = 00000001
RLF f,1
f = 00000010
RLF f,1
f =00000100
RLF f,1
f = 00001000
RLF f,1
f = 00010000

وهكذا

التعليمة

RRF

Rotate left through carry

الصيغة اللغوية:

RRF f,d

ووظيفتها:

تدوير مسجل الهدف(مقدار f واحدة bit) واذاكانت carry flag جهه اليمين عن طريق) تساوى صفر فأن الناتج فى مسجل d( ) واذا كانت تساوى واحد فأن w العمل (الناتج يوضع داخل مسجل الهدف.

مثال:

اذاكان f = 00000000

RRF f,1

f =10000000

RRF f,1

f = 01000000

RRF f,1

f = 00100000

RRF f,1

f = 00010000

RRF f,1

f = 00001000

RRF f,1

وهكذا

التعليمة

INCFSZ

increment f ,skip if 0

الصيغة اللغوية :

تساوى الصفر فأن الناتجdاذا كانت ال
d) واذا كانت الwيوضع داخل مسجل العمل (
) سوف تصبحfتساوى واحد فأن مسجل الهدف(
) .f+1(
) تساوىfاما اذا كانت قيمة مسجل الهدف (
NOPالصفر فأن التعليمة التالية ينفذ بدلآ منها
وغير ذلك ينفذ سير البرنامج طبيعى

مثال:

```
Count   equ     0x5B
MOVLW       0xfB
MOVWF       Count
  loop          INCFSZ    Count,1
GOTO       loop
```

بقية البرنامج؛

التعليمة

DECFSZ

decrement f ,skip if 0

الصيغة اللغوية :

تساوى الصفر فأن الناتجdاذا كانت ال
d) واذا كانت الwيوضع داخل مسجل العمل (
) سوف تصبحfتساوى واحد فأن مسجل الهدف(
) .f-1(
) تساوىfاما اذا كانت قيمة مسجل الهدف (
NOPالصفر فأن التعليمة التالية ينفذ بدلآ منها
وغير ذلك ينفذ سير البرنامج طبيعى

مثال:

```
Count   equ     0x5B
MOVLW       0x03
MOVWF       Count
  loop          DECFSZ    Count,1
GOTO       loop
```

بقية البرنامج؛

التعليمة

GOTO

الصيغة اللغوية :

GOTO k

وضع من قبلLabel عبارة عن kحيث ان الـ

من التعليمات .Bolckالمبرمج ليشير لـ

مثال: Label

```
LoopA    MOVLW   0x05
         MOVWF   PORTA
         GOTO    loopA
```

التعليمة

SLEEP

الصيغة اللغوية :
SLEEP

تقوم هذه التعليمة بوضع المتحكم الدقيق فى حالة (Sleep mode) النوم).
مثال
SLEEP

التعليمة

BTFSS

Bit test f,skip if set

الصيغة اللغوية:

BTFSS f,b

تقوم هذه التعليمة بأختيار الـ bit (b(في مسجل الهدف (f) اذا كانت واحد يقوم بتنفيذ التعليمة التالية NOP بدلا من تنفيذ التعليمة واذا كانت صفر فيقوم بتنفيذها.

مثال:

سنقوم باختبار RA0 في الـ PORTA

```
BITISNOTSET   BTFRSS     PORTA,RA0
                GOTO       BITISNOTSET
              GOTO       BITISSET
```

التعليمة

BTFSC

Bit test f,skip if clear

الصيغة اللغوية:

BTFSC f,b

ووظيفتها:

تقوم هذه التعليمة بأختيار الـbit (b) (اذا كانت صفر يقومfفى مسجل الهدف) بتنفيذ التعليمة التاليةNOP بدلا من التعليمة واذا كانت واحد فيقوم بتنفيذها.

مثال:

```
BITISSET   BTFRSC     PORTA,RA0
             GOTO       BITISSET
            GOTO       BITISNOTSET
```

التعليمة

ANDLW

And lietal with w

الصيغة اللغوية :
ADDLW k

ووظيفتها :
من محتوى مسجلAndعمل عمليات
) والناتج يوضعk) الى (wالعمل(
)wفى مسجل العمل (

مثال:
اذا كان لدينا

```
Count  equ  0x2B
MOVLW   0x5f        ;w = 0101111
MOVWF   0x2B        ;Count=01011111
MOVLW   0xA3        ; w =  10100011
ANDLW   Count       ; _ _ _ _ _ _ _ _
```

w,Count بين؛ ANDالان سوف تتم عملية

w =00000011 الان wفاصبح الـ

التعليمة

ANDWF

And w with f

الصيغة اللغوية :

ANDWF f,d

من مسجلANDوهو يقوم بعمل عملية (w العمل (d) واذا كان الـ f) مع مسجل الهدف يساوى الصفر يوضع الناتج فى مسجل العمل ( ) واذا كانت تساوى واحد فيوضع الناتج فىw( ).fمسجل الهدف (

مثال:

```
w = 0x17                    ;00010111
FSR = 0xC2                  ;11000010
ANDWF      FSR ,1
FSR =0x02                   ;00000010
```

التعليمة

RETFIE

return from interrupt

الصيغة اللغوية :
RETFIE

الوظيفة:
الرجوع من الاجراء الخاص بالمقاطعة وضبط عداد البرنامج على
) فعند حدوث مقاطعة يتم ادخالstack tos(
وبعد انتهاء اجراءstack الحالية الى الـPCقيمة
) عندhالمقاطعة والموجود عند العنوان (٠٠٠٤
وجود مقاطعة.

مثال:
RETFIE

التعليمة

SUBWF

subtract w from f

الصيغة اللغوية:

SUBWF f,d

) منwومعناها هو طرح محتوى مسجل العمل (
تساوى الصفرd) واذاكانت الfمسجل الهدف (
)wاذافسوف يوضع الناتج فى مسجل العمل (
واذاكانت تساوى واحد اذا فأن الناتج سوف
).fيوضع فى مسجل الهدف (

مثال:

W=ox03
F= 0x02
SUBWF F,0
W=0x01
F=0x02

W=ox03
F= 0x02
SUBWF F,1
W=0x03
F=0x03

مثال للتعامل مع المتحكم الصغير
PIC16F84

هذا المثال يقوم بعمل عرض أضاء مكون من ٨ ثماني لدات LEDs و تقوم فكرة البرنامج بأختصار علي عمل أزاحة لمحتوي المنفذ بي PORTB لليمين و عند الوصول للأخر يقوم عمل أزاحة للمسجل PORTB مرة أخري لليسار وهكذا

و يعرض كيفية استخدام التعليمات الأتية :

1 - MOVLW

حرك حرف لمسجل العمل W

2 - MOVWF

حرك ما بداخل مسجل العمل W إلي مسجل الهدف F

3 - BSF

أجعل كل بتات مسجل الهدف F مساوية واحد

4 - BCF

أجعل كل بتات مسجل الهدف F مساوية صفر

5 - RLF

زحزح مسجل الهدف واحد لليمين

6 - RRF

زحزح مسجل الهدف واحد لليسار

7 - CALL

نادي

8 - RETURN

رجوع من أجراء فرعي

9 - DECFSZ

صفر أجعل التعليمةFزود واحد و إذا كانت قيمة و أكمل ما بعدهاNOPالتالية

10 - GOTO

محددLABLEأذهب إلي

11- BTFSS

F في مسجل الهدف Bأختبر ما إذا كانت البت تساوي واحد أم لا

12 - ORG

بداية الكود من العنوان المحدد

13 - TRIS

إليWضع الباترن الموجودة في مسجل العمل TRISB أو TRISAمسجل حالة المنفذ سواء

14 - END

إنهاء البرنامج

ماذا نحتاج للبرمجة؟؟

PICنحتاج الى ثلاثة أشياء لبرمجة الـ

١ - جهاز كمبيوتر

٢- برنامج كمبيوتر يعالج البرنامج الذي تكتبه أي يترجمة إلي ملف ذو أمتداد hex

و البرنامج المستخدم في هذة العملية يسمي

MPLAB
و هو من برمجة الشركى المنتجة للمتحكم الدقيق و يمكن تحميلة من موقع الشركة و هو مجاني و هو عبارة عن مجموعة برامج منها بيئة تطوير متكاملة
IDE(Integated Developmrnt Envirnoment)
لبرامج المتحكم الدقيق
و برنامج
MPASMWIN
و هو البرنامج الذي نستخدمة في ترجمة ملف البرنامج ذو الأمتداد
asm
إلي أمتداد
hex

3 – Pic Programmer

جهاز برمجة المتحكم الدقيق
و عن طريق البرنامج المرفق بهذا المبرمج و بعد توصيلة بالكمبيوتر تتم عملية برمجة(حقن السوفت)
برمجة المتحكم الدقبق و بعد ذلك تستطيع أن تضعة في الدائرة التي كتبت البرنامج من أجلها و تشاهد ما ينفذة البرنامج في الدائرة

البرنامج :

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;Author : Mohamed.M.abdelbary
;Date : 07/05/2004
;Version: 1.0
;Title: Led fiesta
;Description: 8 leds fiesta
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

            list  p=16f84a
            include     "p16f84a.inc"
            __CONFIG   _CP_OFF & _WDT_OFF & _PWRTE_OFF & _XT_OSC

G1          equ         h'0f'
G2          equ         h'1f'

            org         0

            bsf     STATUS,RP0
            movlw b'00000000'
            movwf TRISB
            bcf   STATUS,RP0
            movlw b'00000001'
            movwf PORTB
Gloop rlf         PORTB

            Call  Timer



            btfss PORTB,7
            goto  Gloop

Aloop rrf         PORTB


            Call  Timer


            btfss PORTB,0
            goto  Aloop


Timer movlw       d'200'
      movwf       G1
Gloop1      movlw       d'255'
      movwf       G2
Gloop2      decfsz            G2
      goto        Gloop2
      decfsz            G1
      goto        Gloop1
```

```
      return

      end
```

شرح البرنامج :

.:1:.

```
      list  p=16f84a
      include     "p16f84a.inc"
```

وظيفة هذان السطران هي الأشارة إلي أن المتحكم pic16f84الدقيق المستخدم هو
و أن يلجاء إلي الملف المسمي p16f84a.inc
و يأخذ منة الثوابت المكتوبة في البرنامج مثل الثابت PORTA
• فقيمتة هي x05
ففي الملف السابق تكون مكتوبة بالصيغة

```
PORTA       equ   0x05
```

و هكذا مع جميع الثوابت

.:2:.

```
__CONFIG   _CP_OFF & _WDT_OFF & _PWRTE_OFF & _XT_OSC
```

هذا السطر يسمي Configuration Word
أي كلمة ضبط الأعدادات
و هنا نريد أخبار المترجم أن يقوم بأبطال عمل حماية الكود من القرأء Code Protection (CP_OFF)
و مصدر نبضات الساعة الخارجي هو كرستالة كوارتز XT_OSC)

Watch Dog Timerو أبطال عمل الـ
و عادتا ما نريد أستخدام هذا العداد لأن وظيفتة الأساسية هي ضمان عمل المتحكم بدون أن يهنج و هذا في البرامج الكبيرة و لكن في حالة برنامجنا لسنا بحاجة لأن نشغل بالنا بة الأن

.:3:.

```
G1          equ         h'0f'
G2          equ         h'1f'
```

في العناوينG2 , G1هذان السطران يقوما بحجز المتغيران equالمذكرة أمامهم بأستخدام التعليمة

.:4:.

```
org         0
```

يشير هذا السطر أنة سوف تكون بداية الكود عند العنوان رقم ٠ من ذاكرة البرنامج
Program memory

.:5:.

```
bsf     STATUS,RP0
```

تفيد هذة التعليمة في الأنتقال إلأي البانك وان
Bank1
للتعامل مع المسجلات الموجودة في و هنا نريد التعامل مع المسجل المسمي
TRISB
كما في التعليمة التالية

.:6:.

```
tris  PORTB
```

تقوم هذة التعليمة بنقل محتويات مسجل العمل إلي المسجل المسئول عن تحديد أرجل الدخل و الخرج في منفذ معين و هو في مثالنا
PORTB
و مبدأياً في بداية البرنامج تكون محتويات مسجل العمل تساوي الصفر فيكون جميع أرجل المنفذ
PORTB
خرج

.:7:.

```
movlw b'00000001'
```

تحريك هذا البايت إلي مسجل العمل

.:8:.

```
bcf          STATUS,RP0
```

الرجوع مرة أخري للبانك صفر للتعامل مع مسجلاتة

.:9:.

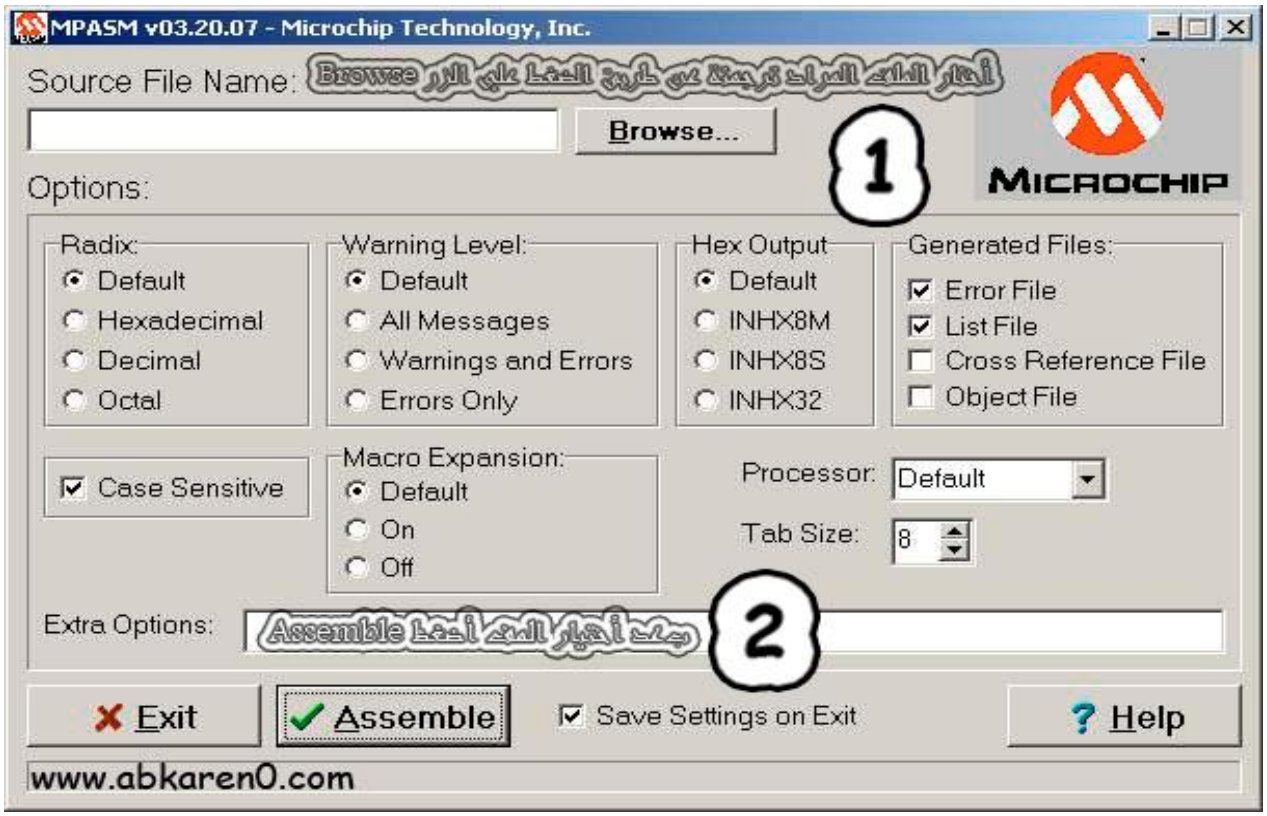
```
movwf PORTB
```

حرك ما بداخل مسجل العمل إلي مسجل الهدف و المقصود هنا هوPORTB

ترجمة البرنامج :

Mirochipقم بفتح البرنامج الخاص بشركة لترجمة برامج المتحكم الصغير

أول شئ قم بأختيار الملف البرنامج الذي قمة Browseبكتابتة عن طريق الضغط علي الزر



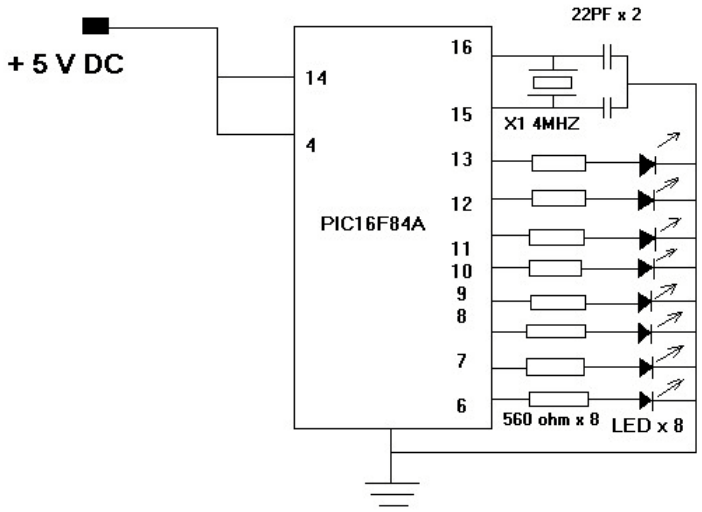
Assembleثاني شئ قم بالضغط علي الزر و أنتظر حتي تظهر نيجة الترجمة و هي بأذن اللّه النجاح كما تري Assembly Succeddful

بقي شئ واحد الأن و هو تنفيذ المشروع Hardware
في خطوتين :
1 - شحن البرنامج المترجم داخل المتحكم الصغير عن طريق مبرمج مناسب.
2 - تنفيذ الدائرة العملية للبرنامج السابق.

الدائرة :

مكونات الدائرة :

عدد	
١	IC PIC 16F84A
١	كريستالة ٤ ميجا هيرتز
٨	مقاومة ٥٦٠ أوم
٨	LED ضوئي
٢	مكثف ٢٢ بيكو فاراد



التحويل العملي للدائرة لأسنخدام برنامج PCB Expreess